Home Appliance Control System

Project Phase I

Team:
Tech-9

Website:
http://ooad.ersoft.org

Team Members:
Abhishek Anand
Abhinav Devpura
Chaitanya Halbe
Eric Anderson
Jamie Smith
LeQiao
Madhura Bhatawdekar
Sandip Gaikwad
Vineet Singh

Table of Contents

1. Introduction

Home appliance control systems (HACS) are not a new concept. They have existed in science fiction but also in implementation. X10 was the first such system and was created in 1975. Although it and other HACS have been available for over 30 years, they have not seen widespread adoption and remain only as a subject of science fiction to the general community. It is believed that the lack of adoption is due to the lack of extensibility, cost, and ease of existing systems.

The Home Appliance Control System (HACS) is developed to control various home appliances such as a Microwave, Oven, Air Conditioner, TV, etc., through one or more controllers such as a cell phone or PDA. In centralized controlled systems, home appliances will be connected to base station, which is called the Home Appliance Controller (HAC), installed in the house. Through HAC we can control all the home appliances by issuing commands from the mobile device. If the control is distributed home appliances will not be connected to base station, rather any appliance will be able to take control over other appliances. HACS will give us easy control over the home appliances even when we are away.

1.1. Scope

Tech-9 hopes to develop a HACS that corrects the listed insufficiencies of previous systems. This document's purpose is to show the development of such a system using some ideas presented by the WRSPM Model. This document's sections are modeled after the World, Requirements, and Specification of the model. Due to Tech-9's limited resources, this document does not provide a complete view of each, but instead just aims to provide just enough understanding of each to allow some traceability to the architecture and design of the system.

2. References

Web references:

http://www.utdallas.edu/~chung/OOAD/syllabus.htm
http://www.utdallas.edu/~weiminma/public/applications.htm
http://www.utdallas.edu/~chung/Fujitsu/UML_2.0/Rumbaugh--UML_2.0_Reference_CD.pdf
http://www.processimpact.com/process_assets/use_case_template.doc

3. Acronyms, and abbreviations

CRC	-	Class Responsibility and Collaboration
HAC	-	Home Appliance Controller
HACS	-	Home Appliance Control System
SD	-	Service Discovery
SNMP	-	Simple Network Management Protocol
UDP	-	User Datagram Protocol

4. Domain Properties

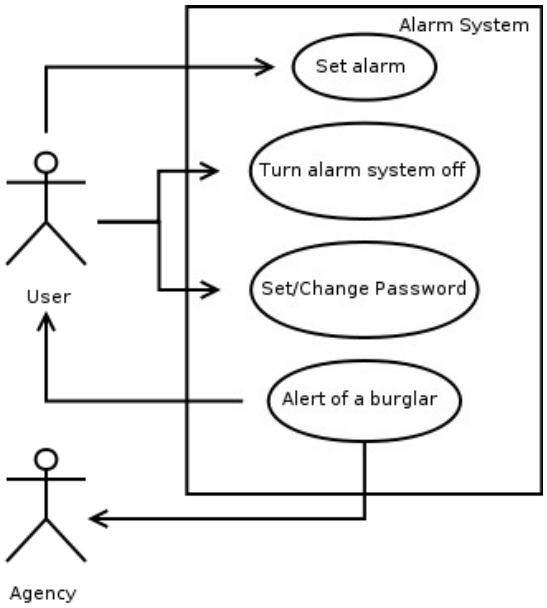
The primary user of the system is an average home owner and his family. This section illustrates how a home owner uses typical appliances within his home. To illustrate this, we will use the examples of an alarm system, a media center, and a standard microwave.

4.1. Use Case Diagrams

4.1.1. Alarm System

A typical Alarm System's goal is to protect the home against intruders.

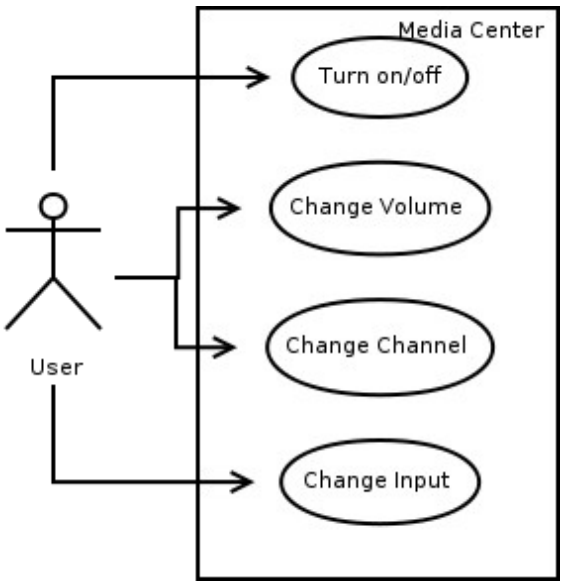
- The User in a typical situation would be the owner of (or someone who lives at) the house. If he is the last person to leave the house, he would set the alarm.
- When the alarm is set, intrusions will sound the alarm if the password is not entered within a certain period of time. Entering the correct password turns the alarm system off.
- The Agency would be the company (or police) that handle the event of an alarm going off.



4.1.2. Media Center

A typical Media Center consists of a television, a receiver with its speakers, and a cable or satellite box.

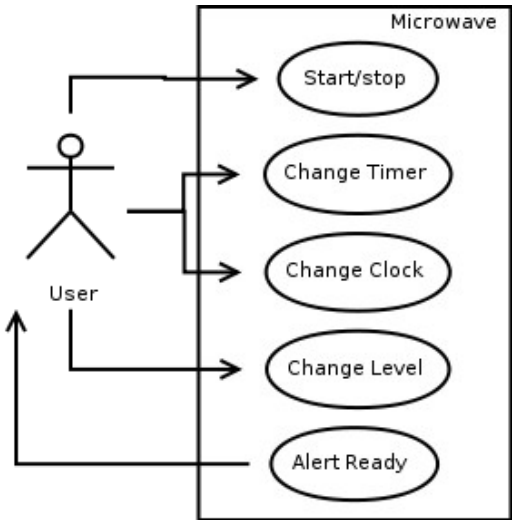
- The User interacts with the media center using a remote control with the capabilities of turning on or off the system, changing the channel, volume and input (i.e. Cable, dvd player, etc).
- A media center will almost certainly have many more functions, however, these are the ones absolutely required for use.



4.1.3. Microwave

A Microwave is a very common home appliance. Studies have implied that some college students would not survive without a microwave.

- The User can set the timer, start and stop the microwave, change the clock, and the heat level.
- Once the timer has reached zero, the Microwave system alerts the user via sound that the timer has reached zero.



4.2. Use Case Templates

These use case templates serve the purpose of defining, for each use case, who is involved, what needs to be done before and after each use case, and the sequence of events for the use case to complete. Due to the simplicity of the devices, many use cases have been combined into one.

4.2.1. Fire Alarm

Use Case Name:	Fire Alarm.
Actors:	User, Agency
Description:	User can set alarm, and change password for the alarm. Agency is notified in case of emergency.
Preconditions:	Set Alarm Password
Post conditions:	Disarm Fire Alarm
Normal Flow:	1. Set Fire Alarm
Alternative Flows:	Notify Agency if an emergency occurs.
Includes:	
Notes and Issues:	

4.2.2. Media Center

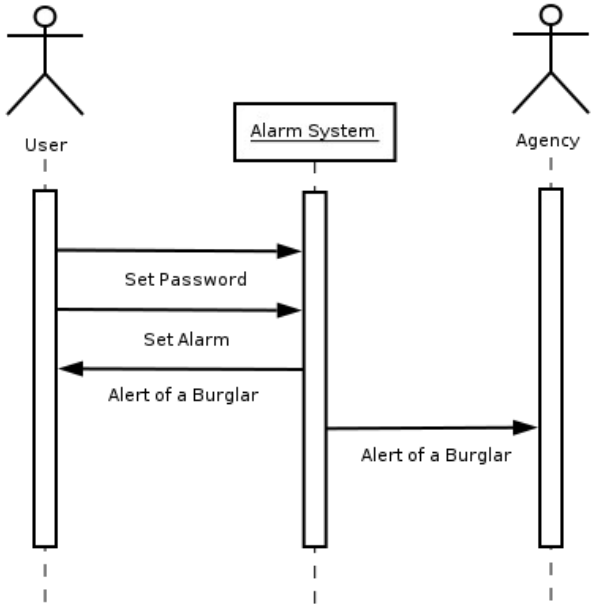
Use Case Name:	Media Center
Actors:	User
Description:	User can turn on/off the system, and change status
Preconditions:	
Post conditions:	
Normal Flow:	1. Turn on Media Center 2. Select Input 3. Change Channel 4. Change Volume 5. Turn off Media Center
Alternative Flows:	
Includes:	
Notes and Issues:	

4.2.3. Microwave

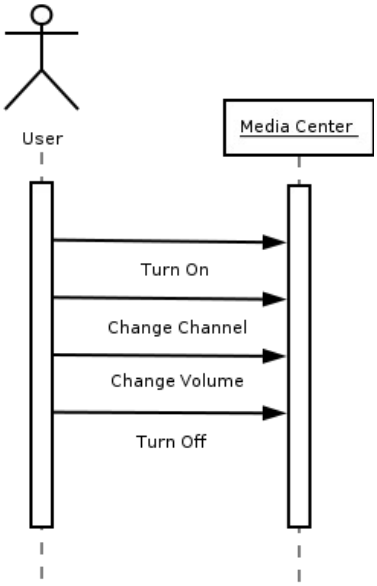
Use Case Name:	Microwave.
Actors:	User
Description:	User can set timer, turn on and off microwave, and set clock and heat level
Preconditions:	
Post conditions:	
Normal Flow:	1. Set Timer 2. Start microwave 3. Microwave alerts user when finished.
Alternative Flows:	
Includes:	
Notes and Issues:	

4.3. Sequence Diagrams

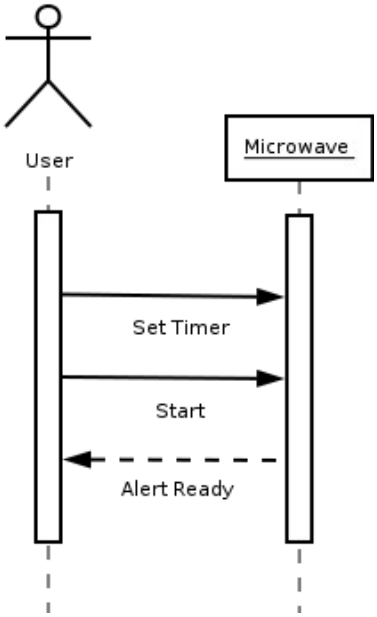
4.3.1. Alarm System



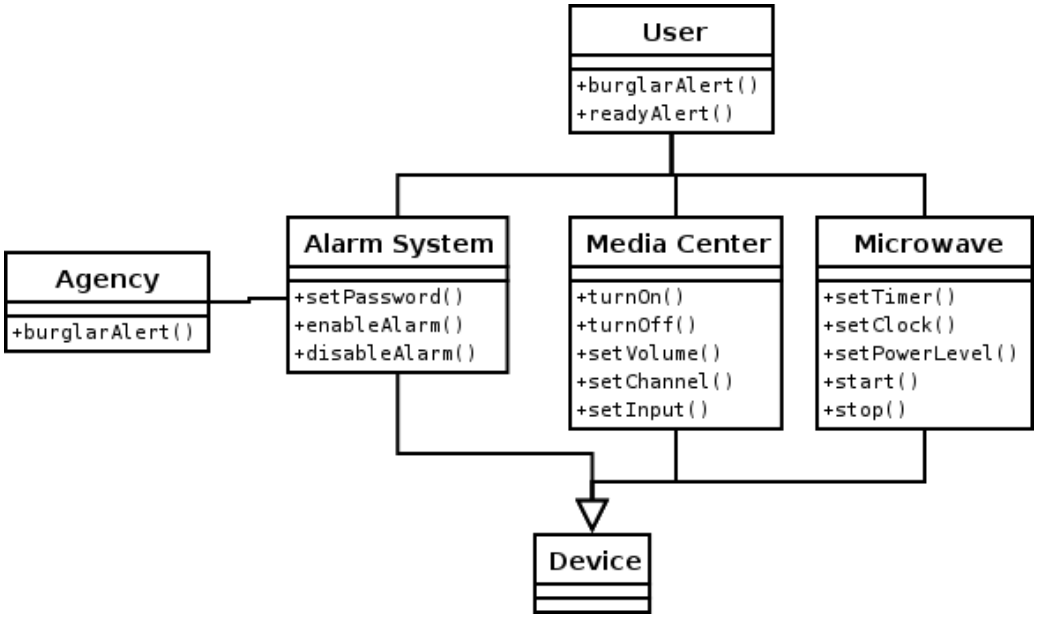
4.3.2. Media Center



4.3.3. Microwave



4.4. Domain Model



5. Requirements

The user needs to interact with or automate the appliances in his home to perform arbitrary tasks. Examples of possible tasks include saving energy by turning out lights; saving energy by adjusting HVAC; finding lost items; controlling appliances when away from home; receiving a notification from an appliance when its filter needs replacement; combining a television, receiver, lights, DVD player, and other peripheral home entertainment devices into a single logical device; and observing the cook time remaining on a stove from a different room. These requirements shall allow the user to perform such tasks. At this point, appliances and devices may be interchanged.

5.1. Functional Requirements

To perform these tasks, the system shall:

- allow devices to be controlled.
- represent devices as a list of properties. Property values must be allowed to be gotten, set, and secured. In addition, notifications for property value changes must be supported.
- provide the user with the ability to browse all the devices connected on the system through a controller. The user can do this remotely when he is away from home. This should give the user all the devices that are in his HACS network.
- allow the devices to be browsable based on device metadata (i.e., device location or device type).
- allow devices to control other devices.
- be able to notify the user when device failures and emergencies occur.
- use a pin code to secure a property. A secured property has either write or read/write security. To write or read/write such a property, you must provide the pin code.
- provide a web interface for away from home use.
- provide basic user authentication for the web interface.
- be able to discover new devices connected to the system.

5.2. Non-functional Requirements

The system shall:

- be installable by an average computer technician.
- be implementable in an appliance with a hardware cost not exceeding 10 dollars.
- require maintenance no more than once a year.
- be repairable by an average computer technician.
- have an intuitive interface. A user shall be able to understand how to interact with a familiar device within 10 minutes of using the system.
- be extensible. The system shall allow arbitrary appliances. The system shall allow third-party interfaces, devices, and enhancements which also shall satisfy these requirements.

The system should:

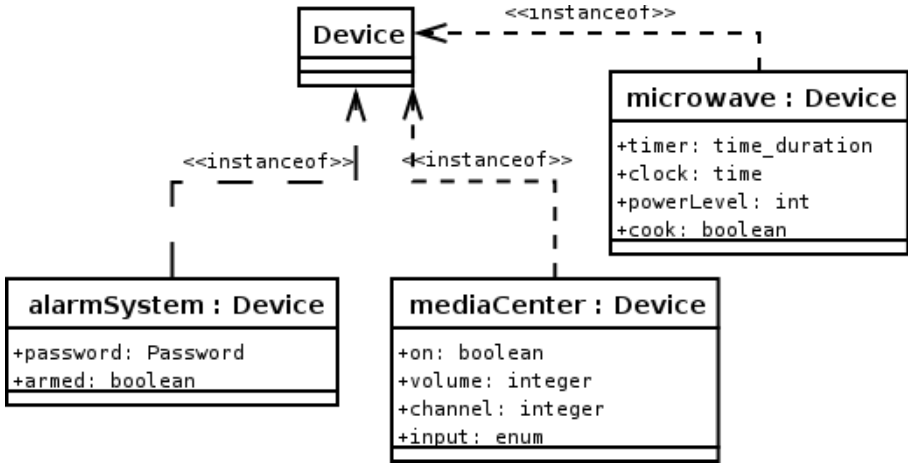
- be installable by an average home user.
- require no configuration after hardware and software installation. This does not imply that advanced features cannot require configuration. It merely states that after the

system's hardware is installed, the appliances should be accessible to the user through the system.

- be repairable by an average home user.

6. Specification

A significant change from the domain model to the specification is that each appliance is no longer a class. Instead, appliances are instances of device with device-specific properties.

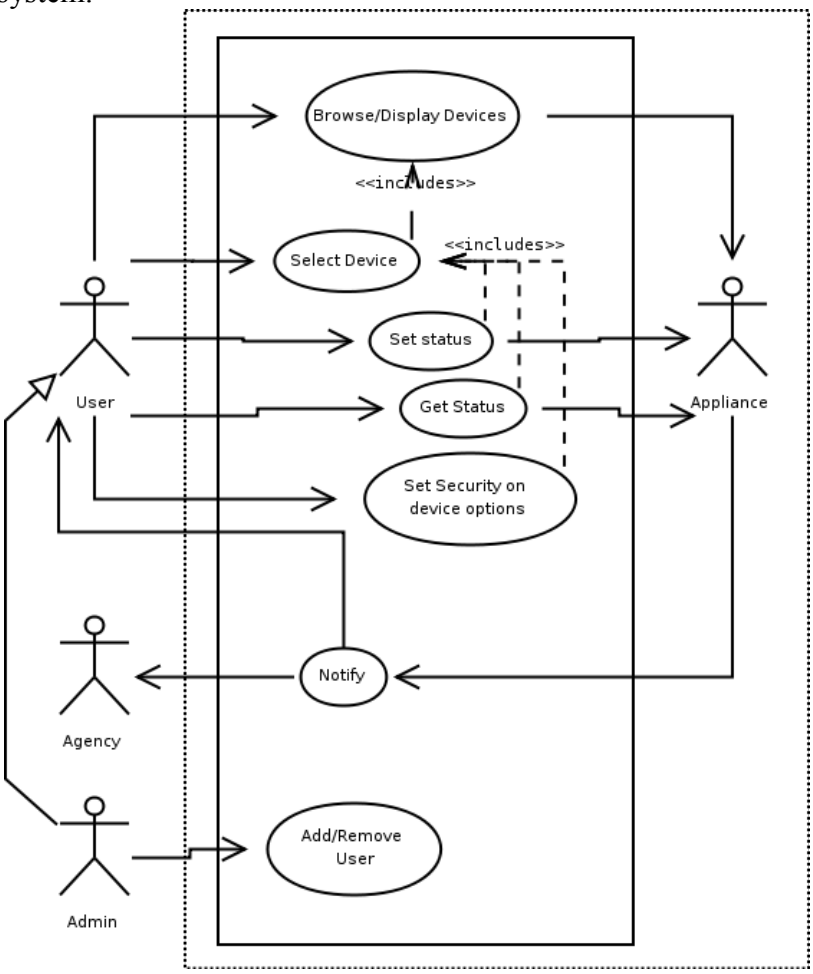


6.1. Use Case Diagram

Overview of the Use Case diagram:

- We have recognized four actors of our system: a general user, an administrator who is a special kind of user, agency and a general actor which depicts all the devices in the house.
- The user can access the browse/display devices use case to have a look at all the devices that are connected in the network of HACS.
- All the appliances sit in the network, which is shown by the dotted boundary in the use case diagram.
- The display device option contains/includes the select device use case. It helps the user to select a device from the list of device the user gets by browsing all the devices.
- After selecting a particular device, the user can get the status of the device, that is, its current working state. He can change the state/ properties of the device using the set status use case.
- The user can also set security on device options, using the set security use case. For example, the parents don't want the children to view some channels. So they can lock that channel (which is a security feature) by entering a 4-digit pin number for that channel.
- All the use cases set status, get status and set security on device options come under one use case which is select device. This is because these use cases can be used only after user selects a device from the network.
- We have a use case notify, which only specific appliances use, if there is some change in their state or if there some failure or emergency. Suppose, there is an accidental fire in the house, so the fir alarm should send a signal to the user warning him that there is an emergency and also this message will be sent to the agency. Then the agency can take appropriate action, in this case it will be alerting the fire department about the fire.

- The administrator is a privileged user, as mentioned earlier, and he has access to all the use cases that a normal user can access and besides this, he can add or remove a user from the system.



6.2. Use Case Templates

6.2.1. Browse/Display Device

Use Case Name:	Browse or Display Device
Actors:	User, Appliances
Description:	Gives list of devices connected to system
Preconditions:	1. All devices should be connected and working.
Post conditions:	
Normal Flow:	4. Log In to system. 5. User Request the list of Devices 6. Controller device sends the discover devices request to Zeroconf. 7. Zeroconf returns the list of devices. 8. List of device will be displayed to user through UI.
Alternative Flows:	
Includes:	Log In.
Notes and Issues:	

6.2.2. Select Device

Use Case Name:	Select Device
Actors:	User
Description:	One of devices connected to system is selected.
Preconditions:	1. All devices should be connected and working.
Post conditions:	
Normal Flow:	1. Discover devices connected to system 2. Give list of devices 3. Select device.
Alternative Flows:	
Includes:	Browse or display devices.
Notes and Issues:	

6.2.3. Set Status

Use Case Name:	Set Device status
Actors:	User, Appliances
Description:	One of parameters of selected device is set.
Preconditions:	1. Selected device should be working.
Post conditions:	
Normal Flow:	1. Select device 2. Choose parameter to be set 3. Set value of parameter.
Alternative Flows:	
Includes:	Select Device.
Notes and Issues:	

6.2.4. Get Status

Use Case Name:	Get Device status
Actors:	User, Appliances
Description:	Get value of parameters of selected device.
Preconditions:	1. Selected device should be working.
Post conditions:	
Normal Flow:	1. Select device 2. Choose parameter to be whose value is needed.
Alternative Flows:	
Includes:	Select Device.
Notes and Issues:	

6.2.5. Set Security on Device Options

Use Case Name:	Set security options.
Actors:	User
Description:	Security options are set for selected device.
Preconditions:	1. Selected device should be working 2. User must have entered correct PIN number.
Post conditions:	
Normal Flow:	1. Select device 2. Enter Pin number. 3. Set security options for device.
Alternative Flows:	
Includes:	Select Device.
Notes and Issues:	

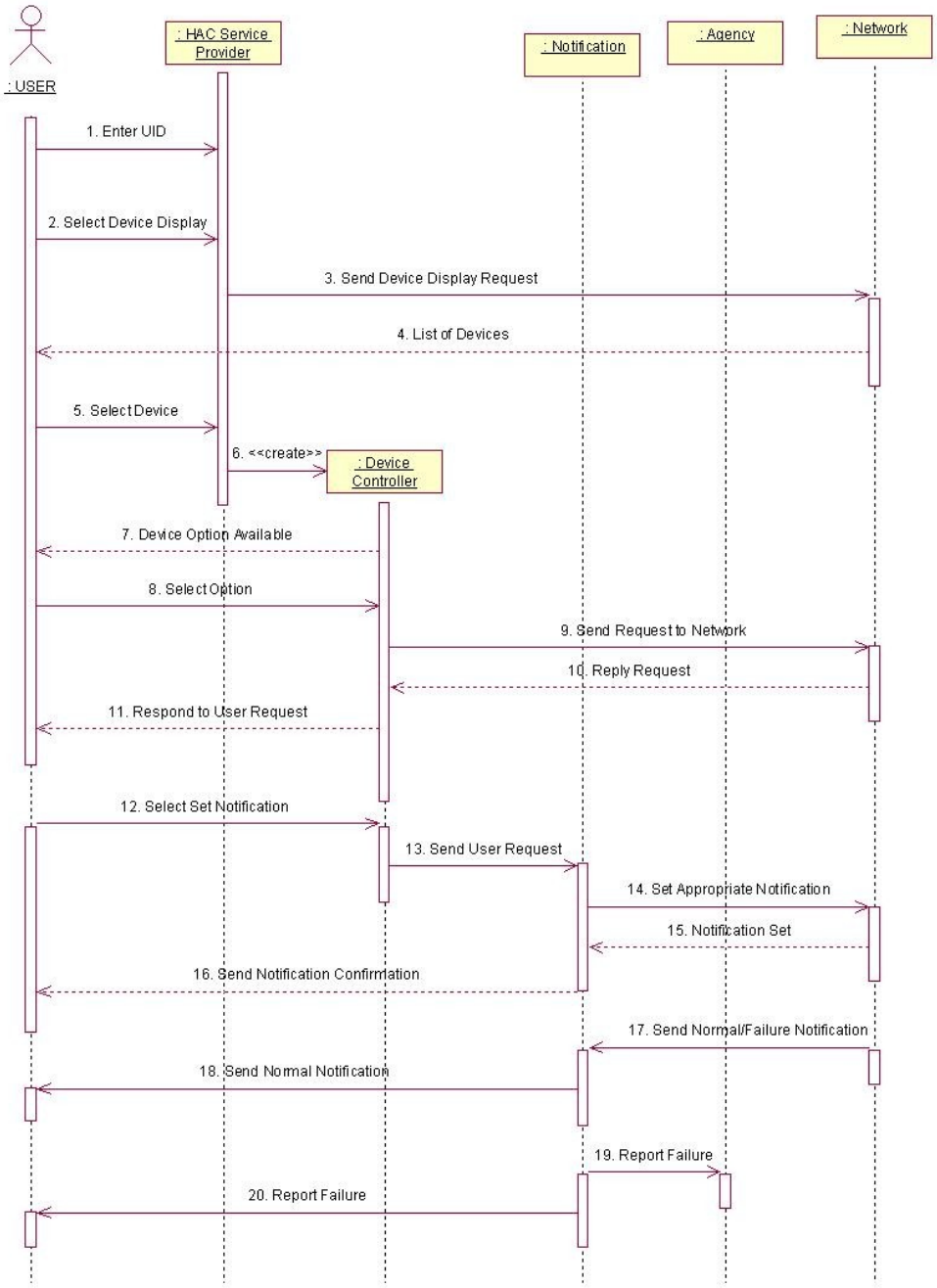
6.2.6. Notify

Use Case Name:	Notify
Actors:	User, Appliances, Agency
Description:	Appliance sends notification to User and to agency in case of any emergency.
Preconditions:	1. User has enabled notification option for device.
Post conditions:	
Normal Flow:	1. Appliance 2. Choose parameter to be set 3. Set value of parameter.
Alternative Flows:	
Includes:	
Notes and Issues:	

6.2.7. Add Remove User

Use Case Name:	Add or Remove device.
Actors:	User, Admin
Description:	Gives access permission to new user.
Preconditions:	
Post conditions:	
Normal Flow:	1. Send request to Admin 2. He will set access permissions for user.
Alternative Flows:	
Includes:	
Notes and Issues:	

6.3. Sequence Diagram



Description:

First of all, user enter user ID, and according to that, if thats a valid user system allows user to enter, acoording to type of user, that is, whether user is normal user or administrator. After

user enters the system, he requests service select device. Then HAC Service Provider send user request to network. Network send all list of devices that are available on network. Then out of those devices, user select particular device which he want to access. By doing that, that particular device is connected to user through HAC Service Provider, and it creates a class, Device Controller on interface, through which user can control device. Then Device Controller send list of options to user of that particular device. For Example, if device is Television, then different options like On/Off or Set volume etc. Out of those options, user select particular option, like if user want to set volume of television, then Device Controller send that request to Network to perform that operation of setting volume. Network reply to request of user to Device Controller and set volume of television accordingly. In turn, Device Controller reply to user about volume set.

There are other options to user like user can select particular notification, like user want to see status of microwave in every 5 mins during cooking. So user can set that notification through Device Controller. Device Controller ask notification class to set that notification, which setup that notification on that particular device and respond to user.

Now, other functions of notification class is to send normal notification to user, like if food is cooked in microwave then notification class will notify user that food is cooked.

Another kind of notification is in time of emergency. For example, if there is fire alert in home then notification class will send emergency notification to user as well as to agency.

6.4. CRC Cards

A CRC Card is used to describe the responsibilities of classes and the interaction between them. The CRC stands for class, responsibilities, and collaborations. It is a very convenient and useful tool to aid with object oriented modeling. Our team will use CRC Cards to present seven classes including appliance, entertainment system, microwave, fire alarm, HACS controller, user, and administrator to illustrate their responsibilities and collaborators.

6.4.1. Appliance

Appliance	
RESPONSIBILTIES	COLLABORATORS
• Transform the commands and results between HACS and particular appliance	HACS Controller Particular Appliance

6.4.2. HACS controller

HACS controller	
RESPONSIBILTIES	COLLABORATORS
• Allow user to log in • Get status • Set status • Manage appliances • Monitor appliances • Add or remove appliances	HACS Controller Particular Appliance User Administrator

6.4.3. User

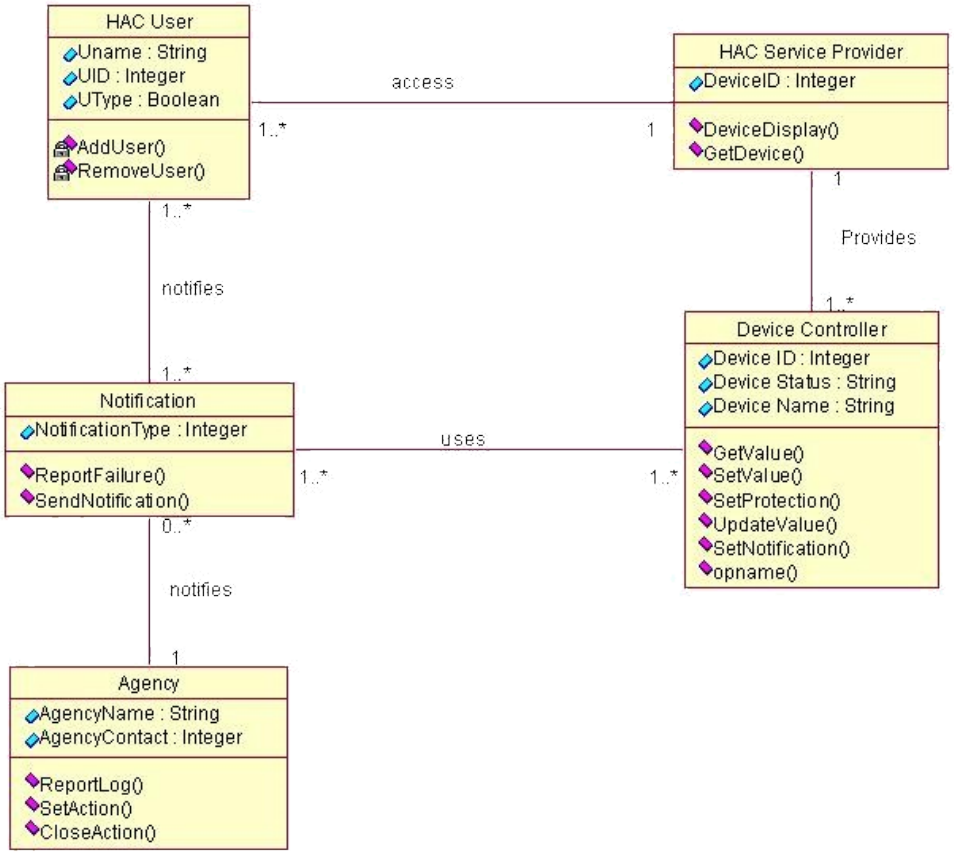
User	
RESPONSIBILTIES	COLLABORATORS
• Log in • Select appliance • Select function • Get status • Log off	HACS Controller

6.4.4. Administrator

Administrator	
RESPONSIBILTIES	COLLABORATORS
• Set permission for user • Higher level control of HACS	HACS Controller User

6.5. Class Diagram

HAC User class tells description about type of user and allows only valid user to enter system in accordance with UID. Now HAC user class can access HAC Service Provider. Now when user will select some kind of device then HAC Service Provider provides Device Controller for that particular device, which will contain attributes and operations of that particular device. Now that controller can use notification class to send notification to user, and also User can set up notification through that Notification class. Now there are different kinds of notification. If it is a normal notification then it is send only to user, and if its some emergency notification, then that notification is send to user as well as agency. Now Agency class has different operations like SetAction and CloseAction, also there will be log of all notifications that come to agency class. Note that Device Controller will manage the properties for each device.



6.5.1. Class – HAC User

Has Attributes:

- Uname – Name of User
- UID – Unique ID to find user
- UType – Differentiates user among administrator and normal users

Has Functions:

- AddUser() – Provides operations to add new system user. Set user type as Admin or Normal user.
- RemoveUser() – Provides operations to remove user from system.

6.5.2. Class – HAC Service Provider

Has Attributes:

- DeviceID – Unique Identifier given to appliances in system.

Has Functions:

- DeviceDisplay() – Creates the list of existing devices and DeviceID.
- GetDevice() – Gives the values of all properties of device after providing DeviceID

6.5.3. Class – Device Controller

Has Attributes:

- DeviceID – Unique Identifier given to appliances in system.
- DeviceStatus – Status of device on/off
- DeviceName – Descriptive name of Device

Has Functions:

- GetValue() – Returns the value of specific property of Device.
- SetValue() – Sets value of specific property of Device.
- SetProtection() – Sets security level of device. Can restrict access to some part of device data.
- UpdateValue() – Sets new value of specific property of Device.
- SetNotification() – Notification operation to user or agency turned on/off.

6.5.4. Class – Notification

Has Attributes:

- NotificationType – Unique Identifier given to appliances in system.

Has Functions:

- ReportFailure() – Reports failure of device to the user.
- SendNotificatioin() – If Notify has been set, Send Notification triggers the alert message to user or agency.

6.5.5. Class – Agency

Has Attributes:

- AgencyName – Unique Identifier given to appliances in system.
- AgencyContact – Phone and Address information of agency

Has Functions:

- ReportLog() – Device and save log of changes in its state. This log can be returned to user on request.
- SetAction() – Provides the action to be taken by agency after specific alert message is received.
- CloseAction – Provides operations to abort or close action taken by agency.

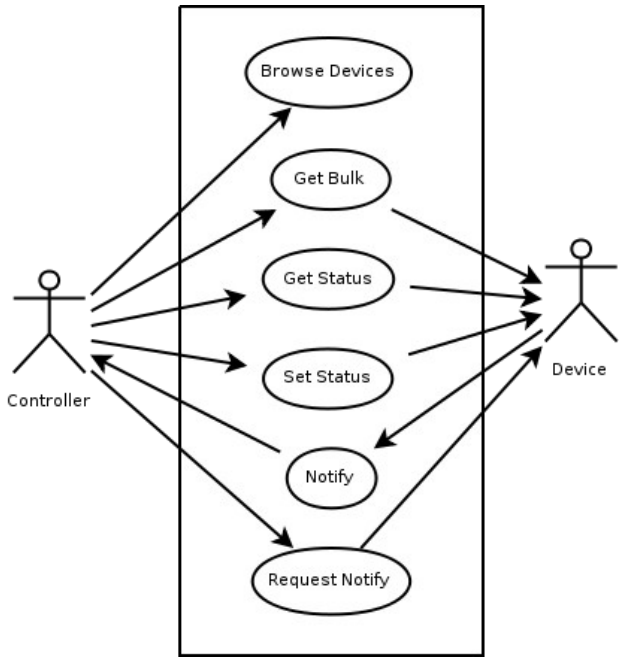
7. Implementation

The system will be divided into two tiers. First tier consists of "Smart Devices", which will be able to take control of any other appliance. Second tier consist of "Dumb Devices". Dumb devices don't have high processing capabilities and will depend on smart devices to receive messages. The communication between devices will have distributed control mostly using smart devices. The system will be capable of discovering and configuring of new appliances brought into home. User will be able to communicate with system through any smart device. Also user will be able to connect remotely to some of smart devices using wireless or web interface.

7.1. Assumptions

1. Smart devices will have sufficient processing power to process, send and receive messages to dumb devices.
2. Dumb devices will have ability to action on messages and provide values of its properties.
3. Distributed system can use wireless or wired networks or both at a time.
4. Wire less network can consist of Infrared, Bluetooth and Cellular network.
5. Preexisting network protocols UDP, SNMP will function same way as expected on both wireless and wired networks.
6. Appliances are capable to support SNMP and Zeroconf (Service Discovery) services.
7. Cellular devices are capable of browsing internet or preloaded with basic appliance control software.

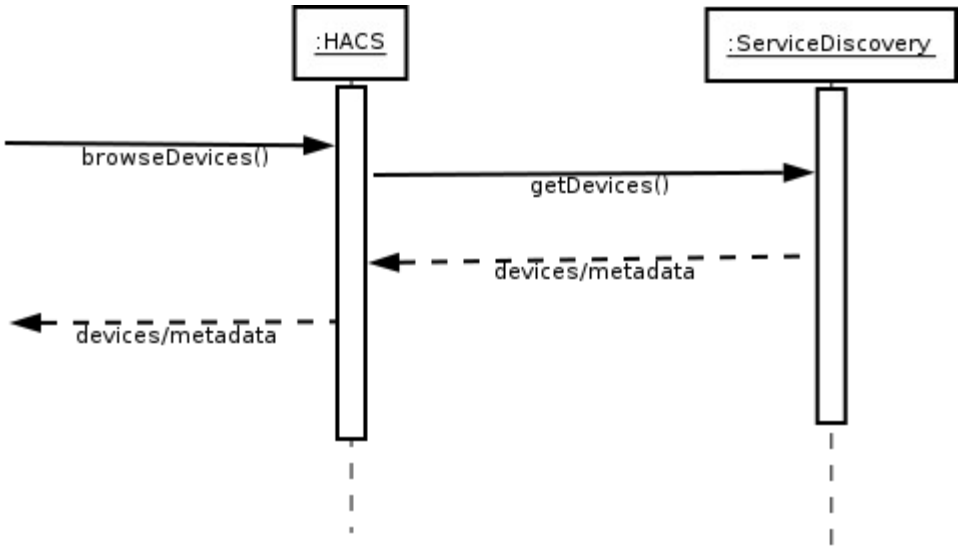
7.2. Use Case Diagram



7.3. Sequence Diagram

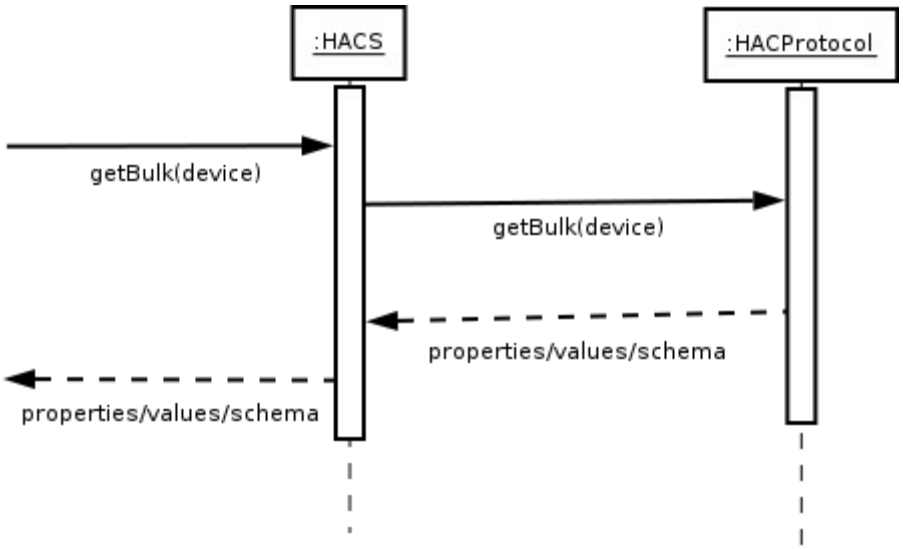
HACS here is the class that the controller uses to perform tasks. Internally, HACS will proxy requests to a ServiceDiscovery protocol and a HACSProtocol. We hope to implement these with Zeroconf and SNMP, respectively.

7.3.1. Browse/Display Device



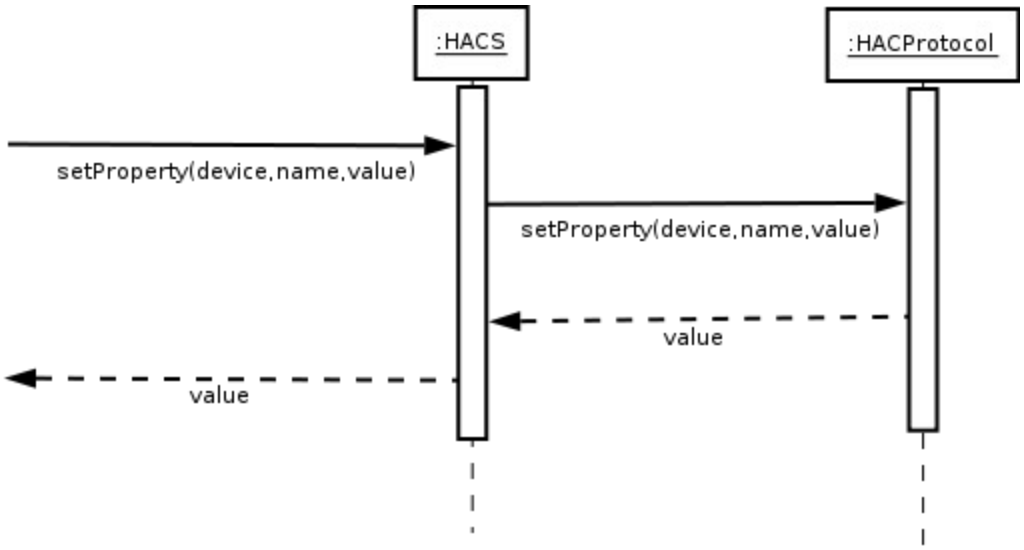
:HACS is an object for communicating with the rest of the HACS. It is being used by the interface. When the user requests the device list, it uses the ServiceDiscovery and gets metadata about all detectable devices. This information is returned to the user through the UI.

7.3.2. Get Status



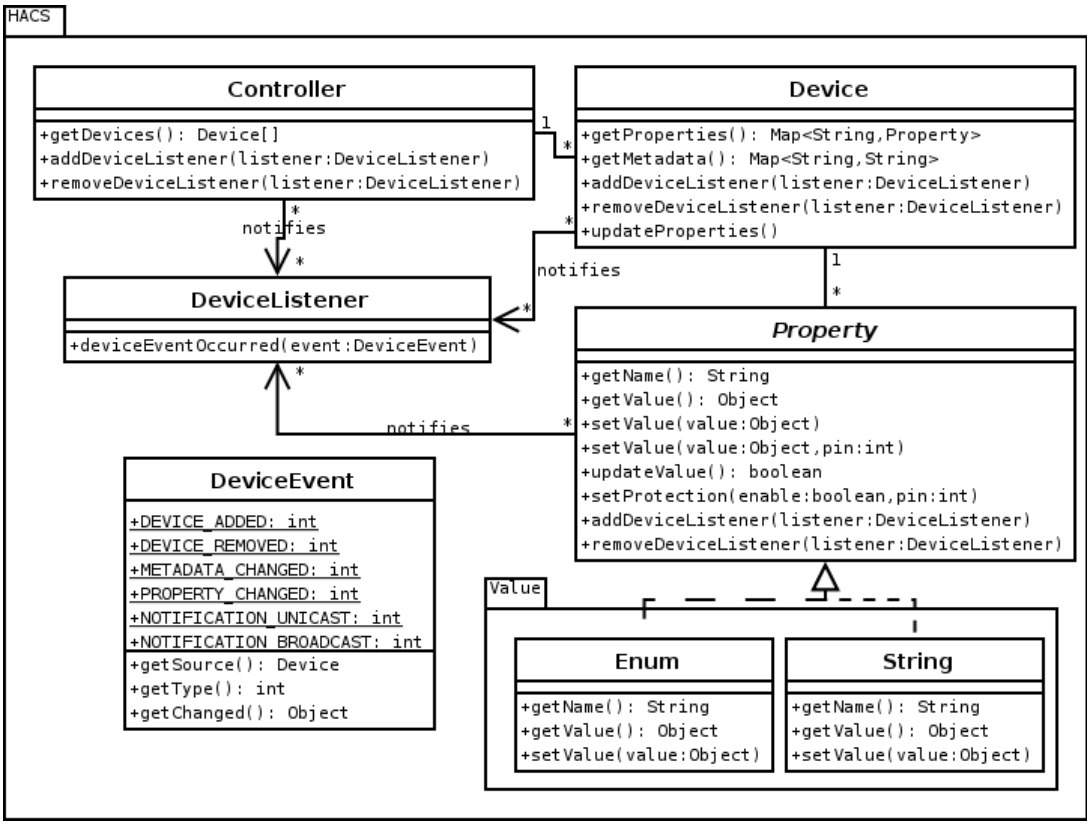
:HACProtocol is an object of the device-controlling protocol. It should be using SNMP. When the user requests status/values of properties of device, the controller communicated with the Smart device using HACProtocol and returns it to user.

7.3.3. Set Status

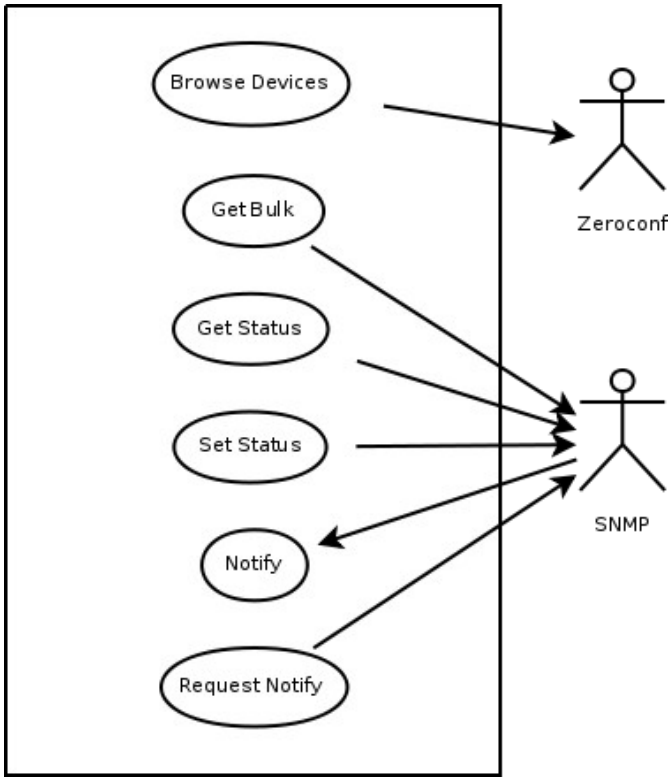


When user sets status/values of properties of device, the controller will request from the Smart device changing the value of a property using HACProtocol. The value of the property is sent back as a response and my be different than requested.

7.4. Class Diagram



7.5. Reusing Existing network functionality.



As we have discussed, we are reusing existing network technology to build simpler system. Above usecases shows how we are using the services of Zeroconf and SNMP to design HACS. HACProtocol is designed on top of SNMP and Zeroconf so that user need not require the implementation details of SNMP and Zeroconf. The existing or new devices are identified in system using services provided by Zeroconf . All other operations on devices are performed using SNMP protocol.